

Appendix C: Utilities

This section describes the utility subprograms supplied with the Business Service Administration subsystem. To invoke these subprograms, you must be in the SYSBIZ library.

The following topics are covered:

- Response Subprogram
 - Business Service Interface Subprogram
 - Conversation Factory Utility
 - Multi-Tasking Verification Utility
 - Log Utilities
-

Response Subprogram

The SPUREPLY subprogram is mainly used by servers to send responses back to a client. The response can be defined as a SYSERR message or a hardcoded text string.

This section covers the following topics:

- Features and Benefits
- Response Length Limitation
- Supported Methods
- Message Protocol
- Call Interface

Features and Benefits

SPUREPLY has the following benefits and features:

- Defines a standard protocol for exchanging messages.
- Enables messages to be multilingual if you define them in SYSERR.
- Performs message substitution of :1::2::3: within SYSERR messages.
- Can send other information in addition to a message.

Response Length Limitation

The maximum supported response length is 5000 bytes.

Supported Methods

SPUREPLY supports the following methods (defined in SPLREPLY). One of these methods must be assigned to the SPAREPLY.METHOD parameter before calling SPUREPLY:

Method	Description
SEND-REPLY	Sends a single message reply, with the End of Conversation option.
SEND-WITHOUT-EOC	Sends a multi-part reply. Use the SEND-REPLY method to send the last message of the reply.
LOOKUP-MESSAGE	Looks up the error message text, but does not send it.
SEND-MESSAGE-ONLY	Sends the message text without the standard protocol information.
SEND-MESSAGE-ONLY-WITHOUT-EOC	Same as SEND-MESSAGE-ONLY, but does not include End of Conversation option.

Message Protocol

All messages sent to the client use the following protocol:

Message	Protocol
SIGNATURE (A6)	MSG111 constant. Defines the structure of the send buffer.
RESPONSE-CODE (N4)	Response code passed to SPUREPLY in SPAREPLY. Successful responses use a response code of zero. Other predefined response codes are: <ul style="list-style-type: none"> ● 001 (Replica ID was not matched) ● 9999 (Natural runtime error)
REPLICA-ID (A32)	Replica ID passed from SPAETB.
BUSINESS-SERVICE (A32)	Passed from SPAETB.
SYSERR-LIBRARY (A8)	Name of the SYSERR library containing the message.
MSG-NR (N4)	Number of the SYSERR message.
MESSAGE (A1/1:V)	Message area of send buffer. In most cases, this area contains a message looked up in SYSERR by SPUREPLY. Additional information can also be passed in this area.

Call Interface

SPUREPLY supports the following interface:

```

PARAMETER USING SPAREPLY /* Specific parameters
PARAMETER                /* The message portion of the send buffer
  01 SPAREPM
    02 INPUT-OUTPUTS
      03 BUFFER-LENGTH (I2)
      03 MSG-BUFFER (A1/1:V)
PARAMETER USING SPAETB   /* Parameters to SPUETB
PARAMETER USING ETBCB    /* Standard Broker control block
PARAMETER USING CDPDA-M  /* Standard message area

```

This section covers the following topics:

- SPAREPLY Data Area
- SPAREPM Data Area

SPAREPLY Data Area

This data area is passed to SPUREPLY. It contains the following data:

Parameter SPAREPLY Library S452		DBID 13000 FNR 1001	
Command			> +
I	T L Name	F Leng	Index/Init/EM/Name/Comment
Top	-----		
*			
*			
*	Data Area Name: SPAREPLY		Function
*	Created on....: Jun 12, 98		=====
*	Created by....: SAG		This data area is passed to
*			SPUREPLY which is used to
*			send a reply back to a client.
*			
*			The reply structure is
*			defined in SPLREP.
1	SPAREPLY		
2	INPUTS		
3	METHOD	I	1 /* See SPLREPLY
3	RESPONSE-CODE	N	4 /* This response, use zero for
*			/* successful response.
3	SYSERR-INFO		
4	MSG-NR	N	4 /* SYSERR Message number
4	SYSERR-LIBRARY	A	8 /* Defaults to SYSBIZ
4	MSG-DATA	A	32 (1:3) /* Subs. values
*			/* May contain *NNNN references
3	TRANSLATE	L	/* Translate character set. If
*			/* currently EBCDIC, message
*			/* will be translated to ASCII
*			/* and vise-versa.
3	EMBEDDED-MSG-INFO		/* This structure is only used
*			/* when the message to be looked
*			/* up is only a portion of the
*			/* data to be sent. In this case
*			/* you must indicate where the
*			/* message is in the send buffer
4	MSG-START	I	2 /* Byte location of start of msg
4	MSG-LENGTH	I	2 /* Total length of message
*			/* portion.

The fields in this data area include:

Field	Description
METHOD (I1)	Indicates whether you want to perform a send with EOC, send without EOC, or just look up the message text. Assign a value from SPLREPLY.
RESPONSE-CODE (N4)	Contains the response code value sent to the client.
MSG-NR (N4)	If a message is looked up in SYSERR, contains the message number.
SYSERR-LIBRARY (A8)	Name of the library in which to look up messages. By default, all messages are looked up in the SYSBIZ library. If this is not true, specify the library name.
MSG-DATA (A32/1:3)	Contains up to three values for substitution into the message. These values replace the :1::2::3: placeholders in the SYSERR message. Substitution values can be looked up in SYSERR by specifying message data in *nnnn format.
TRANSLATE (L)	Indicates whether the message is translated (from EBCDIC to ASCII or vice versa).
EMBEDDED-MSG-INFO	Indicates whether the message is a portion of the data sent. You must specify where the message is in the Send buffer.
MSG-START (I2)	Indicates the starting position of the message portion of the send buffer if the message retrieved from SYSERR only represents a portion of the data to be sent.
MSG-LENGTH (I2)	Indicates the length of the message portion of the send buffer. This field is only required when MSG-START is assigned.

SPAREPM Data Area

This data area is an example of a standard message area that can be passed to SPUREPLY. Use SPAREPM to send messages up to 250 characters in length. After SYSERR messages are looked up, the resulting message text is returned in this parameter. The values in SPAREPLY.MSG-START and SPAREPLY.MSG-LENGTH determine where the message is assigned. If these values are zero, the message is returned, starting at position 1 and continuing to SPAREPM.BUFFER-LENGTH.

The SPAREPM data area contains the following fields:

Local Command	SPAREPM	Library S452		DBID 13000	FNR 1001	
I T L Name			F Leng	Index/Init/EM/Name/Comment		> +
All	-----					
*	Data Area Name: SPAREPM			Function		
*	Created on....: Jun 12, 98			=====		
*	Created by....: SAG			This data area can be used as		
*				the second parameter to		
*				SPUREPLY. When a message number		
*				is passed to SPUREPLY, the		
*				message text is returned in		
*				this parameter.		
*				Alternatively, the message to		
*				be sent can be passed to		
*				SPUREPLY using this parameter.		
1	SPAREPM					
2	INPUT-OUTPUTS					
3	BUFFER-LENGTH		I	2	INIT<250>	
3	MSG-BUFFER		A	1	(1:250)	
R 3	MSG-BUFFER					
4	MSG-STRING		A	250		

To send information other than a standard message, copy SPAREPM and define the fields you want to send (up to 5000 bytes). To reflect the size of data to be sent, assign the BUFFER-LENGTH field.

Example of a Call

```

/*
/* SYSBIZ/1001: Invalid request:1:sent to:2:expecting:3:
ASSIGN SPAREPLY.MSG-NR = 1001
ASSIGN SPAREPLY.MSG-DATA(1) = #COMMAND
ASSIGN SPAREPLY.MSG-DATA(2) = *PROGRAM
ASSIGN SPAREPLY.MSG-DATA(3) = '''CREATE'''
ASSIGN SPAREPLY.RESPONSE-CODE = 1 /* Invalid command
PERFORM SEND-MESSAGE
*
*****
DEFINE SUBROUTINE SEND-MESSAGE
*****
*
IF #I-AM-ASCII NE #CLIENT-IS-ASCII THEN
  ASSIGN SPAREPLY.TRANSLATE = TRUE
END-IF
ASSIGN SPAREPLY.METHOD = SPLREPLY.SEND-REPLY /* Send with eoc
CALLNAT 'SPUREPLY' SPAREPLY
      SPAREPM
      SPAETB
      ETBCB
      MSG-INFO
END-SUBROUTINE /* SEND-MESSAGE

```

Example of Send Buffer

```

MSG1110001ATTACH-MANAGER--B0B218EC55E1AE01 AURORA-CONVERSATION FACTORY
SYSBIZ 1001 Invalid request CMD SH sent to SPSCFACT expecting 'CREATE'

```

where:

- *MSG111* is the message signature
- *0001* is the response code
- *ATTACH-MANAGER--B0B218EC55E1AE01* is the server replica ID
- *AURORA-CONVERSATION-FACTORY* is the Natural Business Services server
- *SYSBIZ* is the name of the SYSERR library used
- *1001* is the SYSERR message number
- *Invalid request CMD SH sent to SPSCFACT expecting 'CREATE'* is the message text

Business Service Interface Subprogram

Writing robust servers can be a complex task. There are many possible errors that can occur, and ensuring that each error is handled in the proper way is very difficult. Some errors are caused by resource shortages, so it is desirable to retry the call again after a brief pause. Other errors are fatal and should result in the server shutting down. Still other errors, like wait timeouts, are normal and expected.

To help simplify and standardize the task of writing servers, Natural Business Services supplies a subprogram that wraps the EntireX ACI calls. This wrapper subprogram, called SPUETB, handles many situations that have to be coded to make direct EntireX calls. To ensure that errors are handled and logged properly, use SPUETB for all EntireX calls.

Features and Benefits

This section contains a summary of the capabilities offered by SPUETB.

The following topics are covered:

- EntireX Error Handling
- Error Logging
- Shutdown Requests
- Server Timeouts
- Command Handling

EntireX Error Handling

Most EntireX errors are handled internally by SPUETB. If the errors are due to resource shortages, SPUETB pauses for two seconds and then tries the call again. The subprogram continues to retry the call for up to 20 seconds.

When implementing server receive loops, SPUETB handles all wait timeouts (EntireX error 74) and returns to the receive state.

Fatal errors cause the server to shut down if SPUETB is granted shutdown permission.

SPUETB can also handle message length errors and return a message to the sender indicating that the message was too long.

Error Logging

All errors returned from EntireX are logged in the Natural Business Services communication log. Use this log to help detect problems with your programs or environment.

Shutdown Requests

SPUETB responds to shutdown requests from EntireX. These requests can be initiated using the EntireX Control Center.

Server Timeouts

Whenever the server has not received a message for the length of time specified on the server record, the server shuts down.

Command Handling

SPUETB registers for the CMD interface and responds to all command requests. Command requests include the CMD CALLNAT command, which allows you to supply the name of the subprogram call.

SPUETB Interface

SPUETB is called using the following interface:

```
DEFINE DATA
  PARAMETER USING SPAETB      /* Specific Parameters
  PARAMETER USING ETBCB      /* Standard Broker control block
  PARAMETER
  01 SEND-BUFFER(A1/1:V)
  01 RECEIVE-BUFFER(A1/1:V)
  01 RESERVED-AREA(A1/1:V) /* Reserved for SPUETB use
  PARAMETER USING CDPDA-M    /* Standard message area
END-DEFINE
```

As in a direct call to EntireX, the caller is responsible for filling in the EntireX control block. Additionally, the caller can specify the degree of error handling and support for common functions handled by SPUETB. This section describes these data areas.

The following topics are covered:

- SPAETB Data Area
- ETBCB Data Area
- SEND-BUFFER
- RECEIVE-BUFFER
- RESERVED-AREA
- CDPDA-M

SPAETB Data Area

The following example shows the SPAETB data area:

```

* Data Area Name: SPAETB                               Function
* Created on.....: May 05, 98                           =====
* Created by....: SAG                                  This data area is passed to
*                                                       SPUETB which is used to
*                                                       encapsulate calls to Entire
*                                                       Broker. Use SPLETB to assign
*                                                       constant values.
*
DEFINE DATA PARAMETER
1 SPAETB
2 FORCE-PDA (A1/1:V) /* This field is only here
*                               /* to force the caller to create
*                               /* a separate LDA to call SPUETB
*                               /* rather than using SPAETB.
*                               /* This way, initial values can
*                               /* be placed in the LDA so that
*                               /* defaults get assigned.
2 INPUTS
3 METHOD (I1) /* 0 = Normal call
*                               /* See SPLETB for other methods
3 ENCAPSULATED-FUNCTIONS /* Set desired functions ...
4 SUPPORT-SERVER-COMMANDS (L) /* SPUETB will automatically
*                               /* register a command service
*                               /* whenever a regular service is
*                               /* registered. CMD is used as
*                               /* the broker service name.
*                               /* CPUETB will handle all
*                               /* command requests directly.
4 ALTER-RECEIVE-SERVICE (L) /* Automatically change the
*                               /* service name on receive to
*                               /* an '*' to allow commands
4 SHUTDOWN-PERMISSION (L) /* If true, SPUETB is allowed to
*                               /* shutdown the server directly.
*                               /* See SHUT-DOWN-REASONS
4 SHUTDOWN-REASONS /* Set desired shutdown reasons:
5 EXPLICIT-SHUTDOWN (L) /* Shutdown request from BROKER
*                               /* or from NBS console.
5 TIMEOUT-REACHED (L) /* See TIMEOUT-HANDLING
5 TERMINAL-ERROR (L) /* Non-recoverable broker error.
4 TIMEOUT-HANDLING (I4) /* 0 = Return all timeouts so
*                               /* that caller can handle
*                               /* >0= Reissue call for this
*                               /* many seconds. Set to
*                               /* max desired idle period.
*                               /* -1= Reissue call indefinitely
*                               /* -1 is normally used by
*                               /* ATTACH servers which
*                               /* should run forever.
3 ERROR-HANDLING
4 HANDLE-TRUNCATION-ERROR (L) /* SPUETB will respond to
*                               /* ETB error 00200094. This
*                               /* won't be sent back to caller
4 ERROR-HANDLER (A8) /* Normally, SPUETB will handle
*                               /* all broker errors. However,
*                               /* if you want to handle these
*                               /* explicitly, supply the
*                               /* routine to be called.
4 USE-BUSINESS-ERROR-LOG (L) /* Log all errors on the NBS
*                               /* file. Warning, this will
*                               /* cause an ET to be issued.
4 WRITE-ERRORS-TO-CONSOLE (L) /* CALL 'CMWTO' with errors
4 WRITE-ERRORS-TO-PRINT-FILE-0 (L) /* Write errors to NATURAL
*                               /* print file 0
4 MAX-RETRY-TIME (I2) /* Number of seconds to continue
*                               /* to retry call in the event of
*                               /* a Broker resource shortage.
*                               /* Defaults to 20 seconds.
4 MESSAGE-DATA /* These fields are used to
*                               /* build helpful error messages
*                               /* when broker calls fail.
5 CALLING-PROGRAM (A8) /* Name of caller.
5 BUSINESS-SERVICE (A32) /* Name of business service
*                               /* if known.
5 CALL-DESC (A32) /* Description of the call
2 INPUT-OUTPUTS
3 REPLIC-IDA (A32) /* Assigned at first LOGON
*                               /* do not adjust
3 CLIENT-MODE (L) /* In this mode, errors need
*                               /* not be logged and checks
*                               /* for broker error cycles
*                               /* are not performed.
3 OPTION (A50) /* SPUETB option
2 OUTPUTS
3 RESULT (I1) /* See SPLETB
*                               /* 0 = Normal request
*                               /* 1 = Attach request
*                               /* 2 = Command request
*                               /* 3 = Timeout
*                               /* 4 = Non-terminal error
*                               /* 5 = Terminal error
*                               /* 6 = Restarting after error.
*                               /* only set after method 6
END-DEFINE

```


The fields in the SPAETB data area include:

Field	Description
FORCE-PDA (A1/1:V)	<p>Due to the number of input settings that must be assigned before calling SPUETB, the preferred method of assigning them is to use supplied LDAs, initialized with common defaults settings. The following LDAs are supplied:</p> <ul style="list-style-type: none"> ● SPAETBC (used by client programs) ● SPAETBS (used by server programs)
METHOD (I1)	Determines the type of processing performed by SPUETB. Assign the method values using one of the constants in SPLETB.METHODS.
SPLETB.NORMAL-CALL	Used for all EntireX calls except LOGON, REGISTER, DEREGISTER, and LOGOFF.
SPLETB.LOGON	Uses the value of SPAETB.BUSINESS-SERVICE to look up the EntireX ID, user ID, and corresponding password with which to log onto EntireX. It also executes the EntireX Logon function.
SPLETB.REGISTER-SERVER	Uses SPAETB.BUSINESS-SERVICE to look up the Broker ID, Server class, Server name, and Broker service name and uses these values to Register with EntireX. If the SUPPORT-SERVER-COMMANDS parameter is set to true, this method also registers an additional interface, CMD, to accept commands.
SPLETB.SHUTDOWN-SERVER	Invokes the EntireX Deregister and Logoff functions. It is only used by servers. Always issue a shutdown request before ending server programs. If you want SPUETB to perform a shut down automatically, assign SHUTDOWN-PERMISSION=TRUE.
SPLETB.LOG-SUPPLIED-ERROR	Requests that an application error be logged by SPUETB. The error must be passed in the MSG-INFO.##MSG field of the CDPDA-M data area. The message is logged to locations specified in the SPAETB.ERROR-HANDLING structure.
SPLETB.LOG-NATURAL-ERROR	Tells SPUETB to log the last Natural error that occurred. It is only called from ON ERROR blocks or error transactions (assign to *ERROR-TA). The error is logged to locations specified in the SPAETB.ERROR-HANDLING structure.

Field	Description
SPLETB.GET-SERVICE-DEFAULTS	<p>Assigns the following fields based on the values established at the time of the initial LOGON method:</p> <ul style="list-style-type: none"> ● ETBCB.BROKER-ID ● ETBCB.SERVER-CLASS ● ETBCB.SERVER-NAME ● ETBCB.SERVICE ● ETBCB.USER-ID ● ETBCB.TOKEN ● ETBCB.SECURITY-TOKEN ● SPAETB.TIMEOUT-HANDLING ● SPAETB.BUSINESS-SERVICE
SPLETB.LOGOFF	Performs an EntireX Logoff function. For other methods, refer to SPLETB.
SUPPORT-SERVER-COMMANDS (L)	Tells SPUETB to automatically support command interface services such as PING, SHUTDOWN, etc. SPUETB automatically registers a separate interface service using CMD as the Broker service name. All command requests are handled by SPUETB; the caller need not code any specific support for commands.
ALTER-RECEIVE-SERVICE (L)	Used in conjunction with SUPPORT-SERVER-COMMANDS. If this field is set to true, SPUETB automatically changes the Broker service name specified in any Receive function to an asterisk (*). This allows the Receive function to be satisfied by either a request for the main interface service or a request for the command interface service.
SHUTDOWN-PERMISSION (L)	If true, SPUETB can shut down the current program. Normally, it is only set for server programs. To determine which events allow SPUETB to shut down the running server, assign the fields in SPAETB.SHUTDOWN. SPUETB always logs any errors prior to shutting down.
EXPLICIT-SHUTDOWN (L)	Allows SPUETB to shut down the server as a result of an explicit SHUTDOWN command.
TIMEOUT-REACHED (L)	Allows SPUETB to shut down the server when the server timeout value is reached. This timeout value is passed in the TIMEOUT-HANDLING parameter and defaulted from the Server Timeout field on the server record.

Field	Description
TERMINAL-ERROR (L)	Allows SPUETB to shut down the server in response to a fatal EntireX error.
TIMEOUT-HANDLING (I4)	<p>Tells SPUETB how to handle timeouts when executing EntireX Receive functions. Can be one of the following:</p> <ul style="list-style-type: none"> ● -1 Execute forever (use SPLETB.NO-TIME-LIMIT) to assign this value. ● 0 Return to the caller after the first receive timeout. ● >0 Execute for this many seconds, then either return to the caller or execute shutdown processing (based on SHUTDOWN-PERMISSION and TIMEOUT-REACHED parameters). <p>This field is derived from the Server Timeout value on the server record. If no server timeout is specified, the following defaults are used:</p> <ul style="list-style-type: none"> ● Servers without attach servers -1 ● Servers with attach servers 1200 (= 20 minutes)
RESERVED (A8)	Reserved for future use.
USE-BUSINESS-ERROR-LOG (L)	Logs all errors to the Natural Business Services log file.
WRITE-ERRORS-TO-CONSOLE (L)	Writes all errors to the operator console.
WRITE-ERRORS-TO-PRINT-FILE 0(L)	Writes all errors to Print file 0.
MAX-RETRY-TIME (I2)	Indicates the length of time to continue trying to execute an EntireX call in the event of a resource shortage. This defaults to 20 seconds.
CALLING-PROGRAM (A8)	Identifies the caller of SPUETB. This name is used when logging error messages.
BUSINESS-SERVICE (A32)	To use the Logon and Register methods of SPUETB, specify the name of the server in this field. Also used when writing error messages.
CALL-DESC (A32)	Free-format description of the call used when logging error messages.
REPLICA-ID(A32)	Replica ID assigned to the server (output field only).

Field	Description
CLIENT-MODE(L)	If this flag is set, SPUETB does not log errors and checks for EntireX error cycles are not performed.
RESULT (I1)	<p>Interpreted after the call to determine the results of the call. The SPLETB data area defines the following constants to check the results:</p> <ul style="list-style-type: none"> ● NORMAL-REQUEST EntireX call completed normally. ● ATTACH-REQUEST EntireX call resulted in an attach request. Only returned to attach servers. ● TIMEOUT Receive timeout was reached and shutdown permission for timeouts was not granted to SPUETB. ● NON-TERMINAL-ERROR Non-terminal EntireX error occurred. This error is automatically logged by SPUETB. See ETBCB.ERROR-CODE. ● TERMINAL-ERROR Terminal EntireX error occurred, but shutdown permission was not granted to SPUETB. The error is automatically logged.

ETBCB Data Area

ETBCB is a standard data area representing the fields that must be passed to EntireX when using the EntireX ACI. The calling program should use ETBCB12 or ETBCB13, depending on the version of the stub in use.

SEND-BUFFER

The send buffer is used in conjunction with the EntireX Send function. The size of this buffer must be greater than or equal to the value of ETBCB.SEND-LEN.

RECEIVE-BUFFER

The Receive buffer is used in conjunction with the EntireX Receive function or blocked Send functions. The size of this buffer must be greater than or equal to the value of ETBCB.RECEIVE-LEN.

RESERVED-AREA

This pass area is reserved for future use. Define and pass the SPAETBP.NOT-USED(*) parameter in place of this parameter.

CDPDA-M

This is a standard message area. Whenever SPUETB encounters a non-recoverable error, it returns with the error text in MSG-INFO.##MSG and MSG-INFO.##RETURN-CODE is assigned “E”.

Use the SPUETB Subprogram

For an example of using SPUETB, refer to the SPSTIMS Timestamp Server example. If you need to do your own character set translation (because your messages contain a mixture of printable and binary data), refer to SPSTIMS2.

CMD TRACE

The TRACE command enables and disables tracing of a running server. This feature is used in conjunction with the CSUDEBI utility. The TRACE command accepts an RID to target the command to a specific replica. There are two separate forms of the TRACE command; the one you choose depends on whether you want to enable or disable tracing.

To enable tracing:

- Use the CMD TRACE LOCATION=*n* [options] command.

To disable tracing:

- Use the CMD TRACE OFF command.

Valid Keywords

Valid trace locations are defined in the CSLDEBUG local data area in SYSCST. The following table shows the trace keywords:


```

DEFINE DATA
  PARAMETER USING SPACALLN /* Standard callnat parameters
  PARAMETER USING SPAREPLY /* Reply message parameters
  PARAMETER
  01 RECEIVE-SEND-BUFFER(A1/1:15000)
END-DEFINE
    
```

The CALLNAT command takes the form:

```
CMD CALLNAT subpname parameter_string
```

where:

- *subpname* is the name of the subprogram you want to CALLNAT
- *parameter_string* is any set of characters to be passed to the specified subprogram using the RECEIVE-SEND-BUFFER

For an example of how to write a new CALLNAT interface subprogram, refer to the SPUCMDT subprogram.

Conversation Factory Utility

Natural Business Services includes a facility called a Conversation Factory. This facility works in conjunction with high-level callnat and message queue APIs to facilitate the simple transfer of data between two platforms. The benefits offered by the Conversation Factory and supporting APIs include:

- Allow communication between a client and server without knowledge of EntireX ACI.
- Allow a conversation to be established between two processes, each acting as clients.
- Support multiple concurrent conversations between the same two participants. For example, the Construct generate server listens for specifications on one conversation and cancels requests on another.
- Are used in conjunction with servers launched from the client to establish a conversation between the client who launched a server and the server itself.

On the server, the Conversation Factory consists of the following four subprograms:

Subprogram	Description
SQUOPEN	Opens a new conversation.
SQUSEND	Sends information from one end of the conversation to the other.
SQURECV	Receives information.
SQUCLOSE	Closes the conversation.

For an example of how to use the Conversation Factory APIs, refer to the SQEXAMPL subprogram.

Multi-Tasking Verification Utility

Use this utility to verify that ADALNK has been configured to be re-entrant and that the Natural batch nucleus that uses it is also re-entrant. A re-entrant Natural nucleus is required to run servers in a batch multi-tasking environment.

To start multiple Natural subtasks, use JCL to run the supplied Natural module, TESTTASK, in batch (as documented in Natural Business Services Installation on Mainframes). If your Natural nucleus is re-entrant, TESTTASK will successfully start Natural subtask sessions that will execute the TESTSTSK program, which will then write trace information to work file 1 showing the execution status of the subtasks. Otherwise, the job that runs TESTTASK will not end and will have to be manually cancelled.

Log Utilities

Natural Business Services supplies several utilities for archiving and deleting log data. Most of the parameters apply to all log archive utilities.

The following log utilities are supplied with Natural Business Services:

- Business Service Log Utilities
- Control Record Log Utilities
- Domain Log Utilities
- Group Log Utilities
- Business Service Definition Log Utilities
- Steplib Log Utilities
- User and Group Log Utilities

Business Service Log Utilities

Utility	Description
BSBLARCP	<p>Allows the log data for business services to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Input End Date Indicates the last LOG date to be archived. ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief). ● Delete After Archive Indicates whether to delete log records after they are archived.
BSBLRESP	<p>Restores data to the business service log file. It uses the entire log data created by the BSBLARCP utility. It also generates a log record of the restore process. This utility has the following input field:</p> <ul style="list-style-type: none"> ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief).

Control Record Log Utilities

Utility	Description
BSCTARCP	<p>Allows the log data for the Natural Business Services control record to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Input End Date Indicates the last LOG date to be archived. ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief). ● Delete After Archive Indicates whether to delete log records after they are archived.
BSCTRESP	<p>Restores data to the control record log file. It uses the entire log data created by the BSCTARCP utility. It also generates a log record of the restore process. This utility has the following input field:</p> <ul style="list-style-type: none"> ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief).

Domain Log Utilities

Utility	Description
BSDOARCP	<p>Allows the log data for Natural Business Services domains to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Input End Date Indicates the last LOG date to be archived. ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief). ● Delete After Archive Indicates whether to delete log records after they are archived.
BSDORESP	<p>Restores data to the domain log file. It uses the entire log data created by the BSDOARCP utility. It also generates a log record of the restore process.</p> <p>This utility has the following input field:</p> <ul style="list-style-type: none"> ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief).

Group Log Utilities

Utility	Description
BSGRARCP	<p>Allows the log data for Natural Business Services groups to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Input End Date Indicates the last LOG date to be archived. ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief). ● Delete After Archive Indicates whether to delete log records after they are archived.
BSGRRESP	<p>Restores data to the group log file. It uses the entire log data created by the BSGRARCP utility. It also generates a log record of the restore process.</p> <p>This utility has the following input field:</p> <ul style="list-style-type: none"> ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief).

Business Service Definition Log Utilities

Utility	Description
BSIFARCP	<p>Allows the log data for business service definitions to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Report Type Indicates which report type to display. To include information related to the interface and method data, enter "F" (full). To only display the log for the business service header information, enter "B" (brief). ● Delete After Archive Indicates whether to delete log records after they are archived.
BSIFRESP	<p>Restores data to the business service definition log file. It uses the entire log data created by the BSIFARCP utility. It also generates a log record of the restore process. This utility has the following input field:</p> <ul style="list-style-type: none"> ● Report Type Indicates which report type to display. To include information related to the interface and method data, enter "F" (full). To only display the log for the business service header information, enter "B" (brief).

Steplib Log Utilities

Utility	Description
BSSDARCP	<p>Allows the log data for EntireX step library chains to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Input End Date Indicates the last LOG date to be archived. ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief). ● Delete After Archive Indicates whether to delete log records after they are archived.
BSSDRESP	<p>Restores data to the EntireX steplib log file. It uses the entire log data created by the BSSDARCP utility. It also generates a log record of the restore process. This utility has the following input field:</p> <ul style="list-style-type: none"> ● Full Report Indicates which details to display. To display full details of all data being logged, enter "F" (full). To only show the main log information, enter "B" (brief).

User and Group Log Utilities

Utility	Description
BSUSARCP	<p>Allows the log data for Natural Business Services users and groups to be archived to a work file and optionally deleted from the log files based on a date. It also generates a log record of the archive process.</p> <p>This utility has the following input fields:</p> <ul style="list-style-type: none"> ● Report Type <p>Indicates which report type to display. To include information related to the interface and method data, enter "F" (full). To only display the log for the business service header information, enter "B" (brief).</p> ● Delete After Archive <p>Indicates whether to delete log records after they are archived.</p>
BSUSRESP	<p>Restores data to the users and groups log file. It uses the entire log data created by the BSUSARCP utility. It also generates a log record of the restore process. This utility has the following input field:</p> <ul style="list-style-type: none"> ● Report Type <p>Indicates which report type to display. To include information related to the interface and method data, enter "F" (full). To only display the log for the user and group information, enter "B" (brief).</p>